

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-342161

(43)Date of publication of application : 29.11.2002

(51)Int.Cl.

G06F 12/08

(21)Application number : 2002-132962

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 08.05.2002

(72)Inventor : BILARDI GIANFRANCO
EKANADHAM KATTAMURI
PATTNAIK PRATAP CHANDRA

(30)Priority

Priority number : 2001 854213

Priority date : 11.05.2001

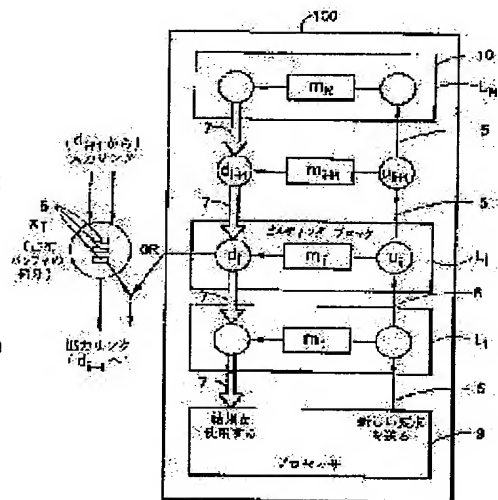
Priority country : US

(54) METHOD FOR TRANSFERRING REQUEST/RESPONSE AND MEMORY STRUCTURE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a memory structure and a method for processing a memory request from processor and returning a response from the various levels of the memory structure to a processor.

SOLUTION: The memory levels L1 to LN of a memory structure 10 are mutually connected by a forward path 5 and a return path 7. The return path has a bandwidth being twice that of the forward path. How many responses are sent from each of the memory levels on the return path to the processor is decided by using an algorithm. The algorithm guarantees a constant bound on the rate of responses sent to the processor. More specifically, if a write request is at the same level which it is targeted, or if a request at a memory level is targeted to a higher memory level, a controller at the memory level on the return path sends two responses to the processor. Otherwise, only one response is sent from the memory level on the return path.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-342161

(P2002-342161A)

(43) 公開日 平成14年11月29日 (2002. 11. 29)

(51) Int.Cl.⁷

G 0 6 F 12/08

識別記号

5 0 9

5 0 1

F I

G 0 6 F 12/08

テーマコード(参考)

5 0 9 B 5 B 0 0 5

5 0 1 Z

審査請求 有 請求項の数 6 O L (全 10 頁)

(21) 出願番号 特願2002-132962(P2002-132962)

(22) 出願日 平成14年 5 月 8 日 (2002. 5. 8)

(31) 優先権主張番号 0 9 / 8 5 4 2 1 3

(32) 優先日 平成13年 5 月 11 日 (2001. 5. 11)

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州
アーモンク ニュー オーチャード ロード

(74) 代理人 100086243

弁理士 坂口 博 (外1名)

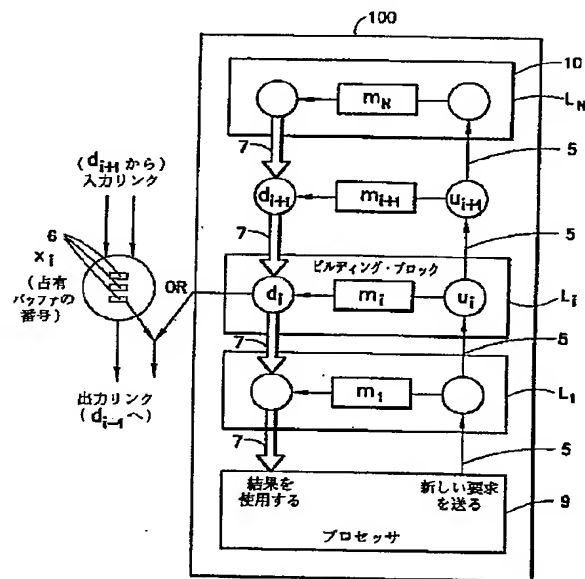
最終頁に続く

(54) 【発明の名称】 要求/応答転送方法およびメモリ構造

(57) 【要約】

【課題】 メモリ構造と、プロセッサからのメモリ要求を処理し、メモリ構造の種々のレベルからプロセッサへ応答を戻す方法とを提供する。

【解決手段】 メモリ構造 10 のメモリレベル $L_1 \sim L_N$ は、フォワード・パス 5 とリターン・パス 7 によって相互接続される。リターン・パスは、フォワード・パスの 2 倍の帯域幅を有する。アルゴリズムを用いて、いかに多くの応答が、リターン・パス上の各メモリ・レベルから、プロセッサへ送られるかを決定する。このアルゴリズムは、プロセッサに送られる応答の速度への一定の束縛を保証する。特に、書込み要求が、ターゲットされる同一のレベルにあるならば、または、メモリ・レベルにおける要求が、高位のメモリ・レベルへターゲットされるならば、リターン・パス上のメモリ・レベルのコントローラから、2 つの応答が送られる。そうでなければ、リターン・パス上のメモリ・レベルから、ただ 1 つの応答が送られる。



【特許請求の範囲】

【請求項1】複数のメモリ・レベルと、前記メモリ・レベルを相互接続するフォワード・パスおよびリターン・パスと、プロセッサとを有するメモリ構造において、前記プロセッサからの要求を、前記フォワード・パス上に転送し、前記要求に対する応答を、前記プロセッサへの前記リターン・パス上に転送する方法であって、前記プロセッサからの各要求を、前記複数のメモリ・レベルの各々を送るステップと、

前記複数のメモリ・レベルのうちの1つのメモリ・レベルからの要求が、前記1つのメモリ・レベルのメモリへの読取り要求であるならば、この読取り要求に対する応答を、前記1つのメモリ・レベルの内部バッファからの他の応答と共に、前記リターン・パス上に送るステップと、

前記複数のメモリ・レベルのうちの1つのメモリ・レベルからの要求が、前記1つのメモリ・レベルのメモリへの書込み要求であるならば、前記1つのメモリ・レベルの内部バッファからの2つの応答を、前記リターン・パス上に送るステップと、

前記複数のメモリ・レベルのうちの1つのメモリ・レベルからの要求が、前記複数のメモリ・レベルのうちの高位のメモリ・レベルをターゲットにしているならば、前記1つのメモリ・レベルの内部バッファからの2つの応答を、前記リターン・パス上に送るステップと、前記複数のメモリ・レベルのうちの1つのメモリ・レベルからの要求が、前記複数のメモリ・レベルのうちの低位のメモリ・レベルをターゲットにしているならば、前記1つのメモリ・レベルの内部バッファからの1つの応答を、前記リターン・パス上に送るステップと、を含む方法。

【請求項2】多くとも3つの応答が、前記複数のメモリ・レベルのうちのいずれかのメモリ・レベルの前記内部バッファに格納される、請求項1に記載の方法。

【請求項3】メモリへの要求を受取り、メモリからの応答を送るメモリ構造において、

メモリ・レベルの順序付きセットを備え、各メモリ・レベルは、コントローラと、前記メモリの一部であるメモリ・ユニットとを有し、

前記メモリ・レベルの低位から始めて、前記メモリ・レベルの高位へ進む、前記メモリ・レベルへ前記要求を送るフォワード・パスと、

前記メモリ・レベルの高位から前記メモリ・レベルの低位へ、前記要求に対する応答を送るリターン・パスとを備え、各メモリ・レベルにおける各コントローラが、以下のアルゴリズムに従って、各メモリ・レベルから低位のメモリ・レベルにおける前記コントローラ内のバッファへのリターン・パスへ、前記要求に対する応答を送るメモリ構造。

a. 前記複数のメモリ・レベルのうちのいずれか1つの

メモリ・レベルにおける要求が、前記1つのメモリ・レベルへの読取り要求であるならば、前記要求に対する第1の応答を、前記1つのメモリ・レベルの前記コントローラの内部バッファからの第2の応答と共に、前記リターン・パス上に送るステップと、

b. 前記複数のメモリ・レベルのうちのいずれか1つのメモリ・レベルにおける要求が、前記1つのメモリ・レベルへの書込み要求であるならば、前記1つのメモリ・レベルの前記コントローラの内部バッファからの2つの応答を、前記1つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上に送るステップと、

c. 前記複数のメモリ・レベルのうちのいずれか1つのメモリ・レベルにおける要求が、前記複数のメモリ・レベルのうちの高位のメモリ・レベルをターゲットとしているならば、前記1つのメモリ・レベルの前記コントローラの内部バッファからの2つの応答を、前記1つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上に送るステップと、

d. 前記複数のメモリ・レベルのうちのいずれか1つのメモリ・レベルにおける要求が、前記複数のメモリ・レベルのうちの低位のメモリ・レベルをターゲットとしているならば、前記1つのメモリ・レベルの前記コントローラの内部バッファからの要求に対する1つの応答を、前記1つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上に送るステップ。

【請求項4】前記リターン・パス上の前記コントローラの各々は、多くとも3つの前記応答を格納するバッファを有する、請求項3に記載のメモリ構造。

【請求項5】前記リターン・パスは、前記フォワード・パスの帯域幅の2倍の帯域幅を有する、請求項3に記載のメモリ構造。

【請求項6】前記複数のメモリ・ユニットの各々は、前記複数のメモリ・ユニットのうちの1つのメモリ・ユニットと、指定された要求との間で、データをコピーし、前記1つの変更された要求を、前記リターン・パス上のコントローラに送る内部ロジックを有する、請求項3に記載のメモリ構造。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一定帯域幅制限下で、限界のないパイプライン深さを支持し、アクセス時間がパイプライン深さに対し線形的にのみ増大する、スケラブル・メモリ・システムに関する。

【0002】

【従来の技術】メモリ技術がプロセッサ技術にスケールせず、およびそれらの速度のミスマッチが存在し続けることはよく知られている。このギャップを埋めるのに用

いられる一般的な手段の1つは、メモリを階層型パイプラインとして設計することである。メモリは、ステージまたはキャッシュのレイヤとして構成される。プロセッサに近いレイヤは、高速であるが、サイズは小さい。一方、プロセッサより遠いレイヤは、低速であり、大きなサイズを有している。離れたステージからのデータの転送の待ち時間を、近いレベルからプロセッサに供給することによって、なくすことができることが望ましい。しかし、現在の構成は、メモリ・サイズが増大するにつれて、適合するようにスケールしていない。メモリの各ステージは異なる構成を有し、したがって待ち時間は非線形であり、予測するのは難しい。高レベルから低レベルへのデータの転送は、メモリ・サブシステムによって暗黙的に管理され、プロセッサはデータ転送については感知しない。転送は、データに対するデマンドによって、あるいは空間局所性のハードウェア・スペキュレーション (speculation) によって、直接にトリガされる。したがって、待ち時間の効果的な減少は、メモリの異なるコンポーネントの故に帯域幅をかなり増大させることなしには、実現できない。

【0003】以上の観点から、以下の特徴を有するスケーラブル・メモリを実現することが望まれる。

1. メモリは、同一のビルディング・ブロックのインスタンスを付加することによって、均一に（あらゆるサイズに対して）構成できなければならない。
2. ビルディング・ブロック（メモリ・レベル）の特性は、技術（物理学の法則にのみ対する科目である）とは、無関係でなければならない。すなわち、ブロック内のメモリは、採用される特定の技術によって指図される有限のサイズおよび速度を有しており、ブロックは、他のブロックからは物理的に分離されており、2個のブロック間の通信には、それらの距離とそれらのレイアウトの許容し得るトポロジーとに基づいて、有限の時間がかかる。
3. メモリ内のロケーションへのアクセスの待ち時間は、プロセッサからの距離に線形比例しなければならない。
4. ブロック間で必要とされる通信帯域幅は、メモリのサイズに関係なく一定でなければならない。
5. メモリは、メモリのサイズに関係なく、一定の速度で要求を受け取り、比例速度で結果を与えることができないなければならない。

【0004】

【発明が解決しようとする課題】本発明の目的は、メモリのロケーションへのアクセス時間が、プロセッサからのメモリ・ロケーションの距離に線形比例するメモリ・システムを提供することにある。

【0005】本発明の他の目的は、メモリの異なるレベル間に必要とされる帯域幅が、メモリのレベルの数とは関係なく、一定である線形コストメモリを提供すること

にある。

【0006】本発明のさらに他の目的は、一定の入力速度で要求を受け取り、入力速度の一定倍率の速度で、これら要求の応答を与えることにある。

【0007】本発明の一般的な目的は、追加のメモリ・レベルを付加することによって、任意のサイズに構成できるスケーラブル・メモリを提供することにある。

【0008】

【課題を解決するための手段】本発明は、メモリ要求を処理するメモリ構造を提供する。このメモリ構造は、多数個のメモリ・レベルを有し、各メモリ・レベルは、メモリ・ユニットと、第1および第2のメモリ・コントローラとを有している。

【0009】メモリ要求は、オペレーション・コード（読取りまたは書込み）と、ターゲット・レベルと、任意のデータとを有している。メモリ要求は、オペレーションが書込みの場合にのみ、データを搬送する。読取りオペレーションの要求に応じて、メモリ応答が発生する。書込みオペレーションに対しては、応答はない。応答は、同様のフォーマットを有している。このフォーマットは、データ・フィールドにデータを有している。

【0010】本発明のメモリ構造は、また、各メモリ要求をメモリ・レベルに送るフォワード・パスと、このフォワード・パスの帯域幅の少なくとも2倍の帯域幅を有するリターン・パスとを有している。

【0011】このリターン・パスは、メモリ要求に対する応答を搬送するために用いられる。第1のメモリ・コントローラは、各メモリ要求のコピーを作って、1つのコピーをメモリ・ユニットに送り、他のコピーを、高レベルの第1のメモリ・コントローラに送る。読取りオペレーションに対しては、メモリ・ユニットは、メモリからのデータを、メモリ要求にコピーする。書込みオペレーションに対しては、メモリ要求からのデータを、メモリにコピーする。これら両方の場合において、メモリ要求のコピーは、そのレベルにおける第2のメモリ・コントローラに送られる。第2のメモリ・コントローラは、メモリ要求を廃棄するか、またはメモリ要求をメモリ応答に変換して、メモリ応答を、以下に説明するように、リターン・パス上を下位レベルの第2のコントローラへ送る。

【0012】a. 読取り要求をメモリ・ユニットから受取り、読取り要求が、読取り要求のターゲット・レベルと同じレベルにあるならば、読取り要求は、応答に変換されて、そのレベルにおける第2のメモリ・コントローラの内部バッファからの1つの応答（もしあれば）と共に、下位レベルに送られる。

【0013】b. 書込み要求をメモリ・ユニットから受取り、書込み要求が、書込み要求のターゲット・レベルと同じレベルにあるならば、書込み要求は廃棄され、そのレベルにおける第2のメモリ・コントローラの内部バ

ッファからの2つの応答（もしあれば）が、下位レベルの第2のメモリ・コントローラに送られる。

【0014】c. 要求をメモリ・ユニットから受取り、要求が、要求のターゲット・レベルより低いレベルにあるならば、要求は廃棄され、2つの応答（もしあれば）が、下位レベルの第2のメモリ・コントローラに送られる。

【0015】d. 要求をメモリ・ユニットから受取り、要求が、要求のターゲット・レベルより高いレベルにあるならば、要求は廃棄され、1つの応答（もしあれば）が、下位レベルの第2のメモリ・コントローラに送られる。

【0016】

【発明の実施の形態】図1に、提案に係るメモリ・システム10の構造を示す。説明を容易にするため、メモリ・システムは、1次元に配列された一連のビルディング・ブロック $L_1 \sim L_n$ を有するように示されている。ビルディング・ブロックは、プロセッサに近い L_1 から始まり、プロセッサから離れるにつれて大きくなるように番号が付されている。各 i 番目のビルディング・ブロック L_i は、メモリ・ユニット m_i と、2個のコントローラ u_i , d_i とを有している。すべてのコントローラ u_i は、単一リンク5によって接続されている。この単一リンクは、プロセッサからメモリセルへ要求を送る“アップワード・パス”を形成する。コントローラ d_i は、“リターン・パス”を形成するリンク対7によって接続され、メモリセルからプロセッサへ応答を送る。この構成は、プロセッサからの単位時間あたり1つの要求を、アップワード・パス上で処理することができるが、プロセッサは、リターンパスに沿って、1単位時間内に2応答までを受け取ることができるように要請される。したがって、この構成は、入力帯域幅の2倍である出力帯域幅を必要とする。また、このメモリ・システムに接続された1個のプロセッサ9が示されている。メモリ要求は、ターゲット・メモリセル・アドレスと、オペレーション（読取り／書込み）と、書込み動作ならばデータとを特定する。サイズ n のメモリに対して、ターゲット・アドレスは、1と n との間の任意の整数である。 n より大きい任意の数を用いて、ノーオペレーション（すなわち、プロセッサは実際の要求をサブミットしなかった）をシミュレートすることができる。

【0017】要求および応答は、また、メモリによってインタープリットされない追加の情報を搬送する。例えば、プロセッサが形式 $op(x, z)$ （ x, z は、メモリ・ロケーションのアドレス）の命令を実行する場合、その意味構造は、メモリのロケーション x からのデータにオペレーション（ op ）を実行し、および結果をメモリのロケーション z に格納することである。この命令に対して、サブミットされるメモリ要求は、形式 $[read, x, no-data, \langle op, z \rangle]$ である。この

要求に対する応答は、形式 $[read, x, data, \langle op, z \rangle]$ であり、ここに $data$ は、ロケーション x から検索された情報である。この応答は、プロセッサへの一連の第2のコントローラによって、リターン・パス上に送り出される。図示の例では、応答がプロセッサによって受け取られると、データに対してオペレーションが実行されて、結果すなわち w を得る。次に、他の要求が、プロセッサから第1のコントローラを経てアップワード・パスに送り出され、結果 w をメモリ・ロケーション z に格納する。この要求のフォーマットは、時々、ロケーション z に値 w を格納することを意味する $[write, z, w, no-info]$ のように見えるかもしれない。

【0018】ターゲット・アドレス i を有する各要求は、アップワード・パス上を伝搬し、そのコピーが、レベル毎のメモリ・ユニットに到達する。メモリ・ユニット m_i は、オペレーションに基づいてデータを読取りまたは格納し、データをコントローラ d_i に送る。書込みオペレーションは、なんらの応答を発生することなく、ここで終了する。他方、読取りオペレーションは、要求を応答に変換させる。この応答は、プロセッサに到達するまで、リターン・パスに沿って伝搬する。コミュニケーション・リンクを伝搬する単位時間およびメモリを読取る単位時間を仮定すると、ターゲット・アドレス i への読取り要求は、伝搬中に輻輳がない場合に、 $(2i+1)$ 個の単位時間をとる。コントローラは、輻輳を処理し、前のセクションで説明した要件が課された要求または応答の流れを保証する。図3において、 u_i , m_i , d_i 用の詳細なロジックは、以下のように指定される。

【0019】各サイクルにおけるオペレーション

図1参照

・コントローラ u_i によって受取られた各要求に関し、1つのコピーがメモリ・ユニット m_i に送られ、他のコピーが、コントローラ u_{i+1} に送られる。一番上のコントローラ（ i が n のとき）では、他のコピーが廃棄される。

・メモリからのデータを、読取りオペレーションのための要求にコピーした後に、あるいは要求からのデータを、書込みオペレーションのためのメモリにコピーした後に、メモリ・ユニット m_i は常に要求をコントローラ d_i に送る。

・図1に示すように、コントローラ d_i は、FIFOキューとして構成される3個の内部バッファ6を有している。各サイクルの始めに、コントローラ d_i は、コントローラ d_{i+1} からの2リンク上に存在する要求に対する応答を、その内部バッファへ転送する。次に、以下のアルゴリズム（図3のフローチャートを参照）を用いて、コントローラ d_{i-1} への2出力リンク上に応答を出す。

【0020】1. メモリ・ユニット m_i からの要求が、ロケーション i への読取りであるならば、その要求は応

答に変換されて、出力リンク上に出される。さらに、コントローラ d_i の内部バッファからの1つの応答（もしあれば）が移動されて、出力リンク上に出される（図3のステップ31参照）。

2. メモリ・ユニット m_i からの要求が、ロケーション i への書込みならば（ステップ32）、あるいは要求が、高位ロケーションをターゲットとするならば（ステップ33）、コントローラ d_i の内部バッファからの2個までの応答（もしあれば）が、移動されて、出力リンク上に出される（ステップ34）。

3. メモリ・ユニット m_i からの要求が、低位ロケーションへのものであるならば（ステップ35）、コントローラ d_i の内部バッファからの1個の応答（もしあれば）が移動されて、出力リンク上に出される（ステップ36）。

【0021】モデルの特性

スケラビリティに対して要求される2つの特性、すなわち各ユニット内の一定のバッファ・サイズと、線形アクセス待ち時間とについて説明する。

【0022】・コントローラ d_i の内部バッファのサイズは、決して3を超えない。図2は、各サイクル後に、コントローラ d_i 内の占有されたバッファの数である x_i の可能な遷移を示す。状態 $x_i = 3$ に対するインバariantは、その状態では多くとも1つの応答が入力ラインを経て到来し、このことが x_i は決して3を超えないことを保証することを表明している。インバariantを示すために、状態 $x_i = 3$ への2つの可能な遷移を検討する。状態 $x_i = 3$ への第1の遷移について考える。この遷移は、最初は $x_i = 2$ であり、コントローラ d_{i+1} からのリンク（図1の7を参照）上の2つの入力アーク（arc）が応答を伝搬し、メモリ・ユニット m_i への要求は、ロケーション（ i 以下）へのものであるときに発生する。このことは、次のサイクルにおいて、コントローラ d_{i+1} からの入力アーク上に、多くとも1つの応答が存在しうることを保証する。これは、以下の理由による。すなわち、メモリ・ユニット m_{i+1} が次のサイクルで、メモリ・ユニット m_i がこのサイクルで処理したのと同じの応答のコピーを処理し、したがってコントローラ d_{i+1} は上記アルゴリズムのケース3を行い、ただ1つの応答を出力する（図3の35および36を参照）からである。次に、状態 $x_i = 3$ への第2の遷移について考える。この遷移は、最初は $x_i = 3$ であって、コントローラ d_{i+1} からの1つの到来要求があって、メモリ・ユニット m_i への要求は、ロケーション（ i 以下）に対するものであるときに発生する。このことは、再び、次のサイクルで、コントローラ d_{i+1} からの入力アーク上に多くとも1つの応答があり得ることを保証している。

・ロケーション i への読取り要求は、 $(4i+1)$ サイクル内に、対応する応答をプロセッサへ戻す。この読取

り要求に対する応答は、 $(i+1)$ サイクル後に、コントローラ d_i に到達する。というのは、そのパス長さが $i+1$ であり、これらパス上では遅延がないからである。コントローラ d_i は、アルゴリズムのケース1を実行するときに、応答を出力アーク上に出す。このアルゴリズムによれば、すべてのバッファは、FIFO順序において空であり、バッファからの少なくとも1つの応答が、各サイクル毎に移動される。したがって、コントローラ d_i からの応答は、プロセッサへ到達する前に、最大 $3i$ 単位の遅延を受ける。したがって、アップワード・パスへ要求を送り、および対応する応答を、プロセッサへのリターン・パスへ送る際的全遅延は、多くとも $4i+1$ である。

【0023】この構成は、各ロケーションでのメモリ・オペレーションの順序を保存し、他方では、異なるロケーション上でのオペレーションの終了の順序が指定されないことが観察された。提案したメモリ・システムは、メモリ・チップに収容される標準的なメモリ・ロジックを用いて実現できる。

【0024】図4は、スケラブル・プロセッサ・システム100の一般的な方式を示している。この方式は、線形メモリ10を用いている。便宜上、このメモリを、読取り要求のための1個のポート3と、書込み要求のための2個のポート4とを有するように拡張した。このことは、単に、各サイクルにおいて2つの結果を出すことのできる線形メモリ10と適合させるために行われる。2個の実行ユニット17は、結果14を使って、書込み要求を独立にライン12上にサブミットする。この構成は、これらライン12上の要求が、互いに干渉しないことを保証する。

【0025】読取り要求は、2個のターゲット・アドレス x および y を有している。読取り要求は、最初に、最初のロケーションに送られ、データを収集し、第2のロケーションへのアップワード・パス上をさらに上方に送られ、データの第2の部分のデータを収集する。結果は、要求されたデータ項目の対を搬送するダウンロード・パス13上を送られる。

【0026】命令フォーマット

プロセッサは、レジスタを有さず、すべてのそのデータに対する線形メモリ階層（最下位レベルは、レジスタを表している）を用いている。命令の一般的な形式は、 $op(x, y, z)$ である。ここに、 x, y, z は、メモリ・ロケーションのアドレスであり、その意味構造は、ロケーション x および y からのデータにオペレーション（ op ）を実行し、結果をロケーション z に格納することである。第2のオペランドは、単項オペレーションに対して省略することができる。しばしば用いられる単項命令は、ロケーション x からのデータをロケーション z にコピーする移動（ x, z ）命令である。

【0027】命令の発行および実行

図4において、プロセッサ9は、命令を発行する1個の発行ユニット19と、2個の実行ユニット17とから構成されている。各実行ユニットは、データが得られると、オペレーションを実行することができる。数個の発行ユニットおよび実行ユニットを、1個のハードウェア・チップ上に、ハードウェアで実現することができる。命令は、2つのフェーズ、すなわち発行フェーズおよび実行フェーズを経る。発行フェーズでは、デュアル・オペランド・フェッチが、線形メモリ10にサブミットされる。オペレーション・コード(`op code`)および宛先情報が、要求に付加され、データと共に戻される。実行フェーズは、メモリがデュアル・オペランド・データを戻すときに、開始する。各実行ユニットは、オペランド・データ14を受け取って、書込み要求をメモリ10にサブミットして、結果を格納する。命令は、プログラムにリストされているのと同じ順序で発行される。メモリは、命令によって指定されるデータ以外のデータ移動を行わないので、プログラムはメモリを完全にコントロールし、移動命令を発行することにより、メモリ管理を明確に行って、データが必要とされるときに、データをプロセッサの近くに持ってくる。メモリのパイプライン化された性質は、他の命令を実行しながら、データの同時移動を容易にする。典型的には、本発明を用いて、プログラマは、移動命令を用いて、データがプログラムによって必要とされるときまでに、データをプロセッサの近くに持ってくる。

【0028】読取り—後—書込み障害(Reader-after-write Hazards)

発行ユニットおよび実行ユニットは、同時に動作するので、ロケーションへの書込みオペレーションを、メモリにサブミットすることを保証しなければならない。これは、そのロケーションへの対応する読取りオペレーションがサブミットされる前に行われる。説明のための図2を参照されたい。命令`op(x, y, z)`に対して、`z`への書込み要求は、その命令の発行と実行との間の時間間隔の間は、未解決であると言われる。以下に説明するように、メモリの領域への未解決の書込みを追跡することによって、障害を避けられる。図5において、実線で示すパスは、移動命令21がどのようにして実行されるかを示している。移動命令が発行されると、ロケーション`y`が読取られ、データが実行ユニット17へ送られる。命令が実行されると、結果が、ロケーション`x`に送られて格納される。しかし、発行ユニットは同時に進行して、それに続く他の命令を発行する。加算命令22は、ロケーション`x`を用いる次の命令の例であり、そのパスは、破線で示されている。もし、この加算命令が、ロケーション`x`への前の格納が行われる前に発行されるならば、障害が発生する。

【0029】分離(Fence)カウンタ

プロセッサは、有限組の分離カウンタを備えている。各

分離カウンタは、メモリの領域に関連している。メモリ領域のサイズを2の累乗として選ぶことによって、 $\log N$ を必要とし、したがって、 $\log N$ 個のカウンタの維持は、実際には重大な制限ではない。図6において、`op(x, y, z)`のような命令が発行される(ステップ31)毎に、ロケーション`z`を含む範囲に関連した分離カウンタは、カウントをインクリメントしており(ステップ36)、要求がメモリに送られて、ロケーション、例えば`x`および`y`のデータを読取る。図6のステップ37を参照されたい。例えば、検索されたデータ(`data1, data2`)は、次に、図6のステップ38に示すように、実行ユニットに送られる。実行ユニットが、オペレーション、すなわち`op(data1, data2, z)`を終了し、書込み要求をロケーション`z`にサブミットする(ステップ33)。対応する分離カウンタ(図4の18を参照)は、カウントをデクリメントしている(ステップ34)。したがって、分離カウンタは、その関連する範囲内で、ロケーションへの未解決書込みの数を含む。`fence(region of memory)`の形式で、特殊な命令を挿入することによって、分離は、プログラム(アラ・リリース(`ala release`)整合性モデル)により明確に実行される。図5の23を参照されたい。分離命令が発行ユニットによって遭遇されると(ステップ32)、対応する分離カウンタが零になるまでに、発行ユニットは停止する(ステップ35)。このメカニズムは、読取り/書込み障害を避けるプログラムによって用いることができ、分離オペレーションのための範囲および配置を、明確に選択することによって、有効に行うことができる。典型的に、カウンタは、ハードウェアで実現できる。

【0030】極端なケースは、全オペレーションを、そのターゲット・ロケーション上の分離によって、続行させることである。これは正しく働くが、実行はほぼ順次である。プログラムは、領域に書込む、できるだけ多くの命令を累積するように構成でき、したがって分離命令は、その領域からのロケーションをアクセスする前に通知される。コンパイラ技術を発展させて、位置を明瞭に選択することができる。

【0031】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) 複数のメモリ・レベルと、前記メモリ・レベルを相互接続するフォワード・パスおよびリターン・パスと、プロセッサとを有するメモリ構造において、前記プロセッサからの要求を、前記フォワード・パス上に転送し、前記要求に対する応答を、前記プロセッサへの前記リターン・パス上に転送する方法であって、前記プロセッサからの各要求を、前記複数のメモリ・レベルの各々に送るステップと、前記複数のメモリ・レベルのうちの1つのメモリ・レベルからの要求が、前記1つのメモリ・レベルのメモリへの読取り要求であるならば、この読

取り要求に対する応答を、前記 1 つのメモリ・レベルの内部バッファからの他の応答と共に、前記リターン・パス上を送るステップと、前記複数のメモリ・レベルのうちの 1 つのメモリ・レベルからの要求が、前記 1 つのメモリ・レベルのメモリへの書込み要求であるならば、前記 1 つのメモリ・レベルの内部バッファからの 2 つの応答を、前記リターン・パス上を送るステップと、前記複数のメモリ・レベルのうちの 1 つのメモリ・レベルからの要求が、前記複数のメモリ・レベルのうちの高位のメモリ・レベルをターゲットにしているならば、前記 1 つのメモリ・レベルの内部バッファからの 2 つの応答を、前記リターン・パス上を送るステップと、前記複数のメモリ・レベルのうちの 1 つのメモリ・レベルからの要求が、前記複数のメモリ・レベルのうちの低位のメモリ・レベルをターゲットにしているならば、前記 1 つのメモリ・レベルの内部バッファからの 1 つの応答を、前記リターン・パス上を送るステップと、を含む方法。

(2) 多くとも 3 つの応答が、前記複数のメモリ・レベルのうちのいずれかのメモリ・レベルの前記内部バッファに格納される、上記 (1) に記載の方法。

(3) メモリへの要求を受取り、メモリからの応答を送るメモリ構造において、メモリ・レベルの順序付きセットを備え、各メモリ・レベルは、コントローラと、前記メモリの一部であるメモリ・ユニットとを有し、前記メモリ・レベルの低位から始めて、前記メモリ・レベルの高位へ進む、前記メモリ・レベルへ前記要求を送るフォワード・パスと、前記メモリ・レベルの高位から前記メモリ・レベルの低位へ、前記要求に対する応答を送るリターン・パスとを備え、各メモリ・レベルにおける各コントローラが、以下のアルゴリズムに従って、各メモリ・レベルから低位のメモリ・レベルにおける前記コントローラ内のバッファへのリターン・パスへ、前記要求に対する応答を送るメモリ構造。

a. 前記複数のメモリ・レベルのうちのいずれか 1 つのメモリ・レベルにおける要求が、前記 1 つのメモリ・レベルへの読取り要求であるならば、前記要求に対する第 1 の応答を、前記 1 つのメモリ・レベルの前記コントローラの内部バッファからの第 2 の応答と共に、前記リターン・パス上を送るステップと、

b. 前記複数のメモリ・レベルのうちのいずれか 1 つのメモリ・レベルにおける要求が、前記 1 つのメモリ・レベルへの書込み要求であるならば、前記 1 つのメモリ・レベルの前記コントローラの内部バッファからの 2 つの応答を、前記 1 つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上を送るステップと、

c. 前記複数のメモリ・レベルのうちのいずれか 1 つのメモリ・レベルにおける要求が、前記複数のメモリ・レベルのうちの高位のメモリ・レベルをターゲットとしているならば、前記 1 つのメモリ・レベルの前記コントロ

ーラの内部バッファからの 2 つの応答を、前記 1 つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上を送るステップと、

d. 前記複数のメモリ・レベルのうちのいずれか 1 つのメモリ・レベルにおける要求が、前記複数のメモリ・レベルのうちの低位のメモリ・レベルをターゲットとしているならば、前記 1 つのメモリ・レベルの前記コントローラの内部バッファからの要求に対する 1 つの応答を、前記 1 つのメモリ・レベルよりも低位のメモリ・レベルのコントローラ内のバッファへの前記リターン・パス上を送るステップ。

(4) 前記リターン・パス上の前記コントローラの各々は、多くとも 3 つの前記応答を格納するバッファを有する、上記 (3) に記載のメモリ構造。

(5) 前記リターン・パスは、前記フォワード・パスの帯域幅の 2 倍の帯域幅を有する、上記 (3) に記載のメモリ構造。

(6) 前記複数のメモリ・ユニットの各々は、前記複数のメモリ・ユニットのうちの 1 つのメモリ・ユニットと、指定された要求との間で、データをコピーし、前記 1 つの変更された要求を、前記リターン・パス上のコントローラに送る内部ロジックを有する、上記 (3) に記載のメモリ構造。

【図面の簡単な説明】

【図 1】本発明のメモリ・システムを示す図である。

【図 2】状態マシンの図式的に示す図であり、状態は、プロセッサへのダウンワード・パスに沿って応答を送るコントローラのバッファ・サイズによって定められる。

【図 3】メモリ要求に対する応答を、メモリ・システムのダウンワード・パス上を送る際のコントローラの動作のアルゴリズムを説明するフローチャートである。

【図 4】線形メモリと共に用いることのできるプロセッサを図式的に示す図である。

【図 5】読取り一書込み障害を図式的に示す図である。

【図 6】プロセッサの発行ユニットおよび実行ユニットの動作のフローチャートである。

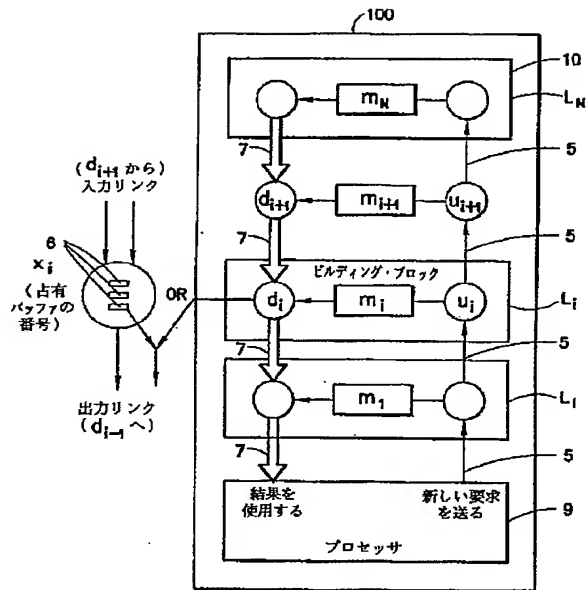
【符号の説明】

- 3 読取り要求のためのポート
- 4 書込み要求のためのポート
- 5 単一リンク
- 6 内部バッファ
- 7 リンク対
- 9 プロセッサ
- 10 メモリ・システム
- 12 ライン
- 13 ダウンロード・パス
- 14 オペランド・データ
- 17 実行ユニット

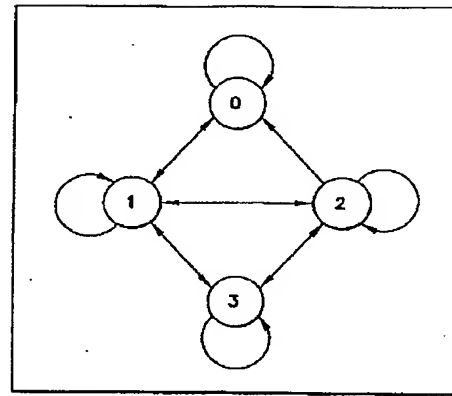
19 発行ユニット
21 移動命令

22 加算命令
100 スケーラブル・プロセッサ・システム

【図1】

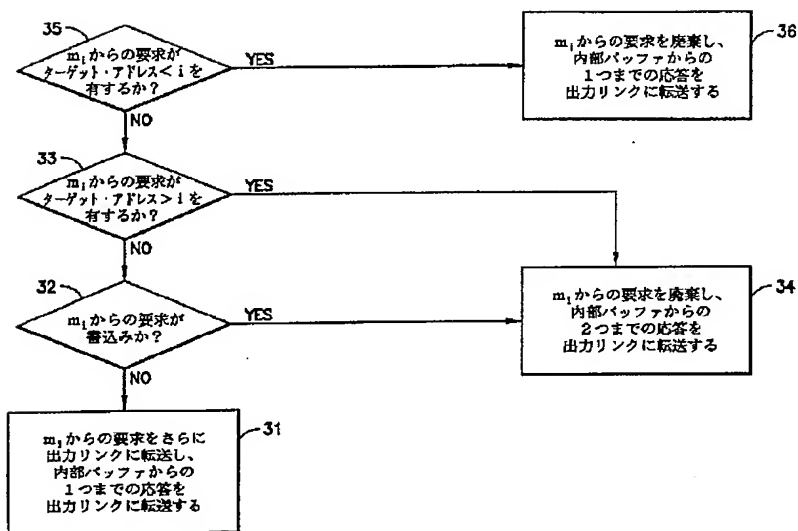


【図2】

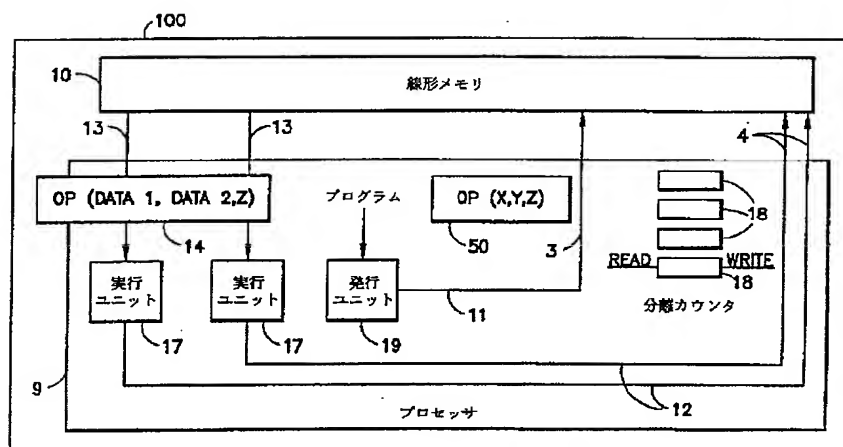


インバリantz:
($X_i=3$) \Rightarrow 入力リンクは、多くとも
1つの要求を有する

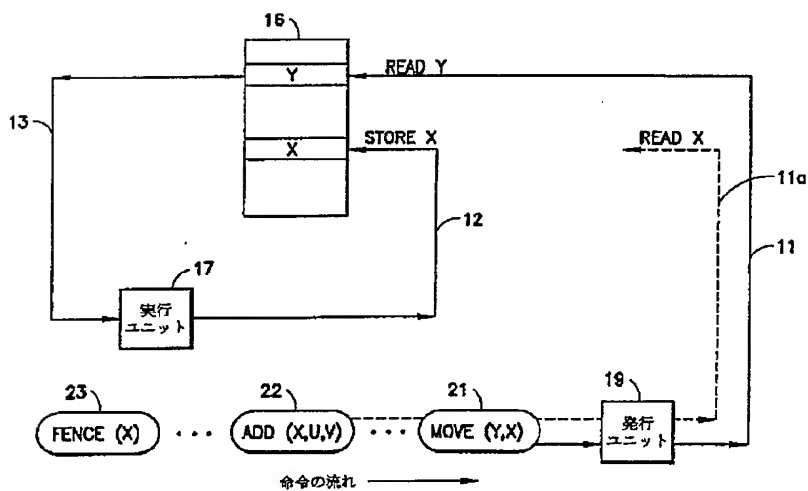
【図3】



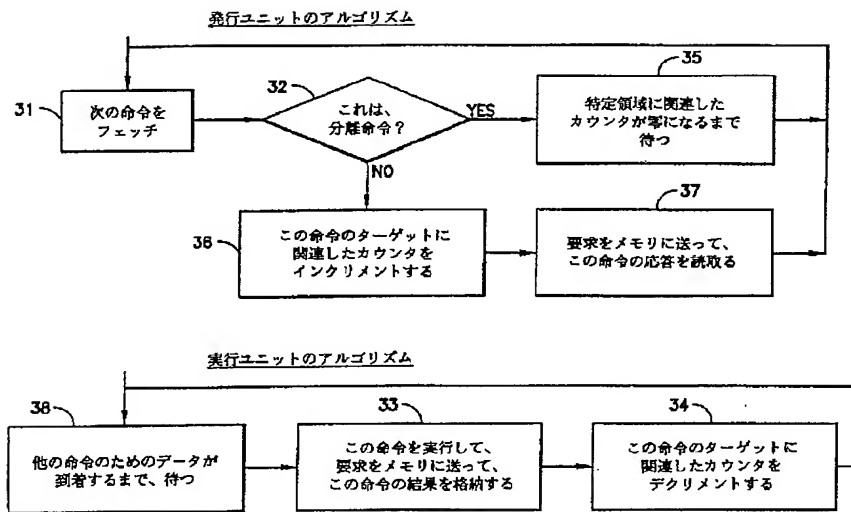
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 ギアンフランコ・ビラルディ
イタリア パドヴァ 35139 ヴィア サ
ン ピエトロ 29

(72)発明者 キッタマリ・エカナダム
アメリカ合衆国 10547 ニューヨーク州
モヒガン レイク プライアーヒル ス
トリート 3855

(72)発明者 プラタップ・チャンドラ・パットネイク
アメリカ合衆国 10562 ニューヨーク州
オッシニング バーネス ストリート

213

Fターム(参考) 5B005 JJ11 MM05 NN12